IN THE SPECIFICATION

Please amend the paragraph beginning on page 12, line 2, and starting with "Information in the sharing vector . . ." as follows:

Information in the sharing vector tracks the location of exclusive or shared copies of a cache line as required to enforce the protocol that maintains coherence between those copies and the home location of the cache line. The sharing vector may be used in one of three ways depending on the directory state. The sharing vector may be in a pointer format as a binary node pointer to a single processor node or input/output node. This format is used when the state is EXCL as well as in most transient states. The sharing vector may be in a pointer timer format as a combination of an input/output read timer and a binary node pointer. This format handles the read exclusive read-only (RDXRO) transaction. The sharing vector may be in a bit vector format as a bit vector of sharers. The field is preferably partitioned into a plane bit vector, a row bit vector, and a column bit vector. This format is used when the cache line is in a SHRD state. Examples of the use of the sharing vector can be found in copending U.S. Application Serial No. 08/971,184 entitled "Multi-dimensional Cache Coherence Directory Structure", now U.S. Patent No. 6,633,958, and in copending U.S. Application Serial No. 09/910,630 entitled "Method and System for Efficient Use of a Multi-dimensional Sharing Vector in a Computer System", both of which are incorporated herein by reference.

Please amend the paragraph beginning on page 47, line 3, and starting with "FIGURE 5 shows . . ." as follows:

FIGURE 5 shows how memory latency can be reduced during read requests. System 10 is a distributed shared memory system with nodes based on snoopy processor buses. When processor 500 makes a read request, a snoop operation is performed at a colocated processor 600 on the local bus. Before the snoop operation is completed, the read request is forwarded from front side bus processor ~~interface 22~~ interface 24 to a local or remote memory directory ~~interface unit 24~~ interface unit 22 for processing. If the snoop operation determines that the cache line needed is held in colocated processor 600 by indicating a processor hit and the data being modified, the data is provided to processor 500 by colocated processor 600 over the local bus for its use. Memory directory ~~interface unit 24~~ interface unit 22 processes the read request and forwards a response to front side bus processor interface 24. Front side bus processor interface 24 sees that the snoop operation satisfied the read request and subsequently discards or ignores the response from memory directory interface unit 22.

Please amend the paragraph beginning on page 49, line 1, and starting with "If the snoop result . . ." as follows:

If the snoop result is that the cache line is not maintained locally or the cache line has not been modified, processor interface 24 forwards the flush request to memory directory ~~interface unit 24~~ interface unit 22 associated with home memory 17 of the cache line. The local processors having an unmodified copy of the cache line may be flushed of the cache line at this point. Memory directory interface unit 22 determines which processors in system 10 maintain a copy of the cache line. The flush request is then forwarded to the identified processors for appropriate action. If an identified processor has a modified copy of the cache line, it removes the modified copy from its cache and forwards the modified copy in a writeback request to memory directory ~~interface unit 24~~ interface unit 22 for memory 17 update.